

5. Frances

Program Name: Frances.java

Input File: frances.dat

Frances's mom works for the local university's academic advising office. As part of her job, she must evaluate degree plans students submit. A degree plan is a student's plan of action for completing a degree and constitutes the order in which he or she plans to take given courses. At the university level, many upper level courses have prerequisites that must be completed. For example, for a computer science major to be able to take a senior level operating systems course, the student must complete a course on data structures and algorithms. But for the student to take data structures and algorithms, the student must take computer science II, but to take computer science II, the student must complete computer science I. Not all courses have prerequisites though. For example, most sophomore level courses can be taken by anyone, at any time. So it doesn't matter if the student takes the class at the very beginning of their college career, at the very end, or somewhere in between.

France's mom sometimes has trouble determining if a given degree plan is in fact valid or not. She has asked Frances to write a program that can verify if a degree plan is legal, i.e. all course prerequisites are met in order and all required courses are taken, or if the degree plan is illegal, i.e. a student plans to take a course without the required prerequisites completed or the student doesn't take all the necessary classes needed to graduate. Can you help Frances write a program that, if given a set of all courses that must be taken as well as the prerequisites present for any given course, determines if the degree plan is in fact legal or not?

Input: The input will consist of an integer T , the number of test cases. T will be in the range of [1,10]. For each test case, input will consist of four lines. Line 1 will contain the names of all the courses a student must take to have a legal degree plan. The number of courses will be greater than or equal to two, and will not exceed forty. Course names are not limited to one-word names. It will be guaranteed that no two courses have the same exact name. Names in the list will be separated by a comma ",". Line 2 will consist of all the course prerequisites. The prerequisites will be given in the form: "Name1->Name2" this means Name1 must be taken before Name2. Prerequisites will be separated by a comma ",". There will be at least one prerequisite present. Line 3 is the order of classes in which the student plans to take the courses. This will be a comma separated list similar to Line 1. Line 4 is 20 dashes and serves to separate all test cases. Note, it is not guaranteed that a course has exactly one prerequisite. It is possible for a course to have more than one prerequisite or zero prerequisites. It is also possible for a course to be the prerequisite of more than one course.

Output: For each test case you are to output: Degree plan #X is legal. or Degree plan #X is illegal. Where X is the degree plan number. Note, for a degree plan to be legal, all courses must be taken, no course can be taken twice, and all prerequisites must be taken in order.

Sample input: (*Lines indented from left are continuation of previous line*)

9

```
CS I,CS II,Introduction to Programming and Problem Solving,Data Structures and
    Algorithms,Operating Systems
CS I->CS II,CS II->Data Structures and Algorithms,Data Structures and Algorithms->Operating
    Systems
CS I,Introduction to Programming and Problem Solving,CS II,Data Structures and
    Algorithms,Operating Systems
-----
CS I,CS II,Introduction to Programming and Problem Solving,Data Structures and
    Algorithms,Operating Systems
CS I->CS II,CS II->Data Structures and Algorithms,Data Structures and Algorithms->Operating
    Systems
CS II,Introduction to Programming and Problem Solving,CS I,Data Structures and
    Algorithms,Operating Systems
-----
CS I,CS II,Introduction to Programming and Problem Solving,Data Structures and
    Algorithms,Operating Systems
CS I->CS II,CS II->Data Structures and Algorithms,Data Structures and Algorithms->Operating
    Systems
CS I,CS II,Data Structures and Algorithms,Operating Systems
-----
CS I,CS II,Introduction to Programming and Problem Solving,Data Structures and
    Algorithms,Operating Systems
```

UIL – Computer Science Programming Packet – State - 2022

```
CS I->CS II,CS II->Data Structures and Algorithms,Data Structures and Algorithms->Operating
Systems
CS I,CS II,Data Structures and Algorithms,Operating Systems,CS I
-----
Course 0,Course 1,Course 2,Course 3,Course 4,Course 5
Course 5->Course 2,Course 5->Course 0,Course 4->Course 0,Course 4->Course 1,Course 2->Course
3,Course 3->Course 1
Course 4,Course 5,Course 2,Course 0,Course 3,Course 1
-----
Course 0,Course 1,Course 2,Course 3,Course 4,Course 5
Course 5->Course 2,Course 5->Course 0,Course 4->Course 0,Course 4->Course 1,Course 2->Course
3,Course 3->Course 1
Course 5,Course 4,Course 2,Course 0,Course 3,Course 1
-----
Course 0,Course 1,Course 2,Course 3,Course 4,Course 5
Course 5->Course 2,Course 5->Course 0,Course 4->Course 0,Course 4->Course 1,Course 2->Course
3,Course 3->Course 1
Course 4,Course 5,Course 2,Course 0,Course 1,Course 3
-----
Course 0,Course 1,Course 2,Course 3,Course 4,Course 5
Course 5->Course 2,Course 5->Course 0,Course 4->Course 0,Course 4->Course 1,Course 2->Course
3,Course 3->Course 1
Course 4,Course 5,Course 2,Course 0,Course 1
-----
Course 0,Course 1,Course 2,Course 3,Course 4,Course 5
Course 0->Course 1,Course 1->Course 2,Course 2->Course 3,Course 3->Course 4,Course 4->Course
5,Course 5->Course 0
Course 0,Course 1,Course 2,Course 3,Course 4,Course 5
-----
```

Sample output:

```
Degree plan #1 is legal.
Degree plan #2 is illegal.
Degree plan #3 is illegal.
Degree plan #4 is illegal.
Degree plan #5 is legal.
Degree plan #6 is legal.
Degree plan #7 is illegal.
Degree plan #8 is illegal.
Degree plan #9 is illegal.
```

Explanation of output:

Degree plan #1: All prerequisites are taken in the correct order and all needed courses are taken.

Degree plan #2: CSII has a prerequisite of CSI, however the student attempted to take CS II before CS I.

Degree plan #3: The student will not take Introduction to Programming and Problem Solving, a required course.

Degree plan #4: The student will not take Introduction to Programming and Problem Solving, a required course. The student attempts to take CS I twice.

Degree plan #5: All prerequisites are taken in the correct order and all needed courses are taken.

Degree plan #6: All prerequisites are taken in the correct order and all needed courses are taken.

Degree plan #7: Course 1 has a prerequisite of Course 3, however the student attempts to take Course 1 before Course 3.

Degree plan #8: The student will not take Course 0, a required course

Degree plan #9: This degree plan is impossible to complete no matter the order the student chooses. This is due to Course 5 having the prerequisite of Course 0.